

# Taller Introducción a la Ingeniería Eléctrica Robótica y comunicaciones basado en Microcontrolador Arduino

Instituto de Ingeniería Eléctrica

Taller 0: Marco del curso e introducción al HW y SW.

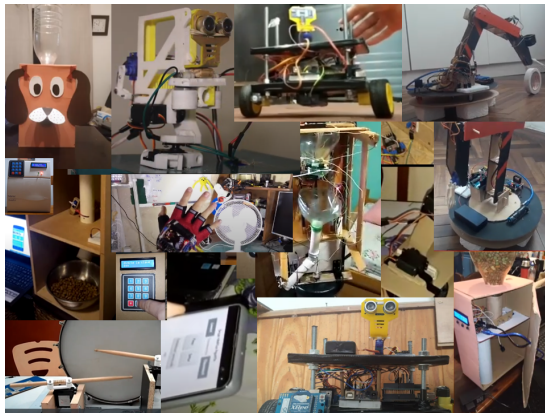
04 de Agosto de 2023

- Objetivos y características.
- Equipo docente.
- Sitio de cursos y foro de consultas.
- Metodología, horario de clases y consulta.
- Cronograma.
- Evaluación.
- Reglas internas básicas.

- Actividad orientada a la generación de ingreso.
- Pretende introducir a la Ingeniería Eléctrica, sus métodos, sus herramientas, sus actores.
  - Página del Instituto de Ingeniería Eléctrica (IIE):  
<https://iie.fing.edu.uy/>
  - Noticias de Instituto de Ingeniería Eléctrica:  
<https://iie.fing.edu.uy/noticias-del-iie/>
- De paso - datos básicos de las carreras:
  - Director de carrera de IE: Julián Oreggioni - [juliano@fing.edu.uy](mailto:juliano@fing.edu.uy) .
  - Directora de carrera de ISC: Claudina Rattaro - [crattaro@fing.edu.uy](mailto:crattaro@fing.edu.uy) .
  - Director de carrera de IFM: Pablo Monzón - [monzon@fing.edu.uy](mailto:monzon@fing.edu.uy).

- Pretende aportar motivación, experiencia de trabajo en equipo, comunicación oral, escrita y digital, a través de una metodología de enseñanza activa.
- **No se viene a escuchar, se viene a HACER!**
- ¿Qué se va a hacer?

- Pretende aportar motivación, experiencia de trabajo en equipo, comunicación oral, escrita y digital, a través de una metodología de enseñanza activa.
- **No se viene a escuchar, se viene a HACER!**
- **¿Qué se va a hacer?**



## Proyectos finales TallerInE Robótico

Primer Semestre 2023

Primer Semestre 2023

Primer Semestre 2021

Primer Semestre 2020

Segundo Semestre 2020

# Objetivos y características

- Se denotan dos etapas bien claras: *Introducción* y *Proyecto final*.
  - **Introducción:** serie de talleres para ir descubriendo el mundo de Arduino. Con énfasis en programación, aplicada al conocimiento y utilización de sensores y actuadores.
  - **Proyecto Final:** cada estudiante, o eventualmente cada grupo de estudiantes, desarrollaran proyectos basados en Arduino, con una tutoría docente "personalizada".
- Para llevar a cabo tanto los ejercicios de la primer etapa, como los proyectos finales, se hará uso de la plataforma Tinkercad, para así acceder a un simulador on-line de Arduino.
- Para la etapa final recibirán materiales, que junto a los conseguidos por cada estudiante, serán utilizados para el armado de los proyectos finales.
- Cuidar el material es responsabilidad de cada estudiante/grupo de estudiantes!!  
(ver las *reglas* en el EVA -Reglas básicas-).

- Se denotan dos etapas bien claras: *Introducción* y *Proyecto final*.
  - **Introducción:** serie de talleres para ir descubriendo el mundo de Arduino. Con énfasis en programación, aplicada al conocimiento y utilización de sensores y actuadores.
  - **Proyecto Final:** cada estudiante, o eventualmente cada grupo de estudiantes, desarrollaran proyectos basados en Arduino, con una tutoría docente "personalizada".
- Para llevar a cabo tanto los ejercicios de la primer etapa, como los proyectos finales, se hará uso de la plataforma Tinkercad, para así acceder a un simulador on-line de Arduino.
- Para la etapa final recibirán materiales, que junto a los conseguidos por cada estudiante, serán utilizados para el armado de los proyectos finales.
- **Cuidar el material es responsabilidad de cada estudiante/grupo de estudiantes!!**  
(ver las *reglas* en el EVA -Reglas básicas-).



## Docentes

- Guillermo Airaldi
- Juan Sánchez
- Pablo Monzón

## Estudiantes Colaboradores

- Santiago Ververis
- Bruno Álvarez
- Gonzalo Chiarlone
- Cristian Criales

- El sitio de cursos: <https://eva.fing.edu.uy>
- Allí se busca el EVA del Tallerine Robótico del segundo semestre dentro de las asignaturas del Instituto de Ingeniería Eléctrica (IIE) (<https://eva.fing.edu.uy/course/view.php?id=712>).
- Dentro del EVA, hay foros de información y foros de consulta.
- Además hay material de apoyo, están los ejercicios a realizar y es la plataforma a través de la cual se realiza buena parte de las evaluaciones.
- Y lo más importante: **es el canal de comunicación oficial del curso.**

- Las clases serán los ***Viernes de 16:00 a 19:00*** horas en el Laboratorio de SW del IIE.
- Serán en un formato que combina presentaciones de temas con mucho trabajo por parte del alumnado.
- Durante la primer etapa, habrá cuestionarios breves e individuales cada semana para afirmar lo incorporado en cada taller.
- Durante la segunda etapa habrá semanas con un esquema de trabajo más libre, ya que trabajarán en grupos y sobre un proyecto.
- Tendremos día semanal de consulta y trabajo, que será los jueves de 14 y las 16 horas, y fijaremos otros.

Cuatro Módulos.

- **Módulo 1:** Introducción a la programación y a Arduino (3-4 talleres).
- **Módulo 2:** Sensores y actuadores (2-3 talleres).
- **Módulo 3:** Comunicaciones (1-2 talleres).
- **Módulo 4:** Proyecto.

# Evaluación:

cuestionarios, ejercicios, participación en clase, proyecto final: entregables y prototipo

- Habrá preguntas para responder individualmente en la semana posterior a cada taller, con el objetivo de obtener una realimentación rápida sobre la asimilación de los temas de parte del alumnado.
- En general, se indicarán lecturas previas a los talleres.
- Se darán ejercicios prácticos domiciliarios a ser compartidos al comienzo del siguiente taller.
- El módulo final tendrá como entregables: un informe, una presentación de apoyo visual y un vídeo.

- Es mandatorio (en una buena :) tener una foto personal y actual en el perfil de la plataforma EVA.
- Ante cualquier duda usar el *Foro de consultas*. Así, se pueden ayudar entre los estudiantes. Además una duda personal, puede cubrir la duda de más estudiantes.
- Recordar que la asistencia a las clases es **obligatoria**.

## 1 Introducción

## 2 Hardware

## 3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:

## 1 Introducción

## 2 Hardware

## 3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:



# ¿Qué es Arduino?

¿Qué es un microcontrolador?

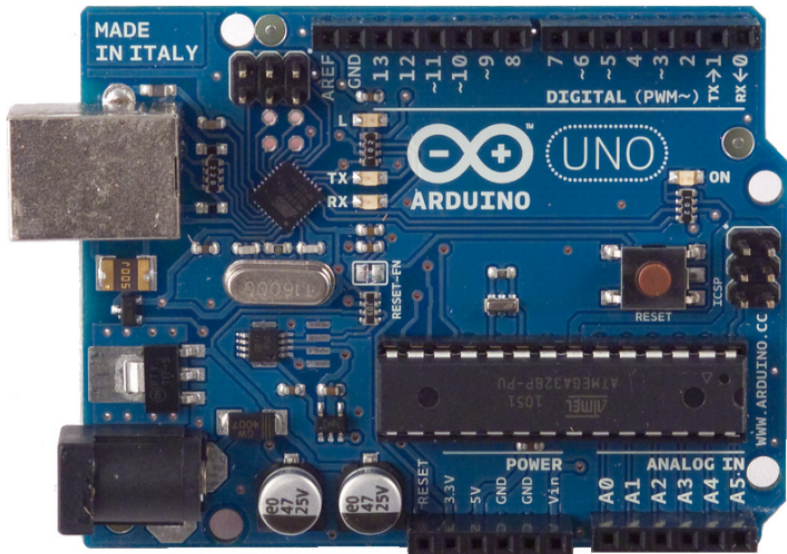
¿Para qué sirve?

# ¿para qué sirve un microcontrolador ( $\mu C$ )?

Algunas de las muchas posibles aplicaciones:

- Robótica.
- Control de un invernadero.
- Control de vehículos.
- Control de un proceso industrial: temperatura, pH, nivel de un tanque, motores, etc.
- Sintetizador de sonidos.
- Domótica.

# Arduino Uno



Basada en el **software/hardware libre**. Permite:

- Estudiar el hardware para entender cómo funciona.
- Hacer modificaciones al hardware.
- Poder compartir esas modificaciones con la comunidad.



## WHAT IS ARDUINO?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

[Learn more about Arduino](#)



## ARDUINO BOARD

Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

[Discover the official Arduino boards](#)



## ARDUINO SOFTWARE

You can tell your Arduino what to do by writing code in the Arduino programming language and using the Arduino development environment.

[Download the Arduino Software](#)

CLOSE

BUY AN ARDUINO

LATEST



## Consecuencias:

- Comunidad activa de usuarios y desarrolladores (foros, ejemplos, tutoriales, etc.).
- Gran cantidad de software y hardware disponible para autoconstruir.
- Precios bajos.
- Existe mucho material en el Web y muchos proyectos interesantes que pueden hacer.

# Esquema de la presentación

## 1 Introducción

## 2 Hardware

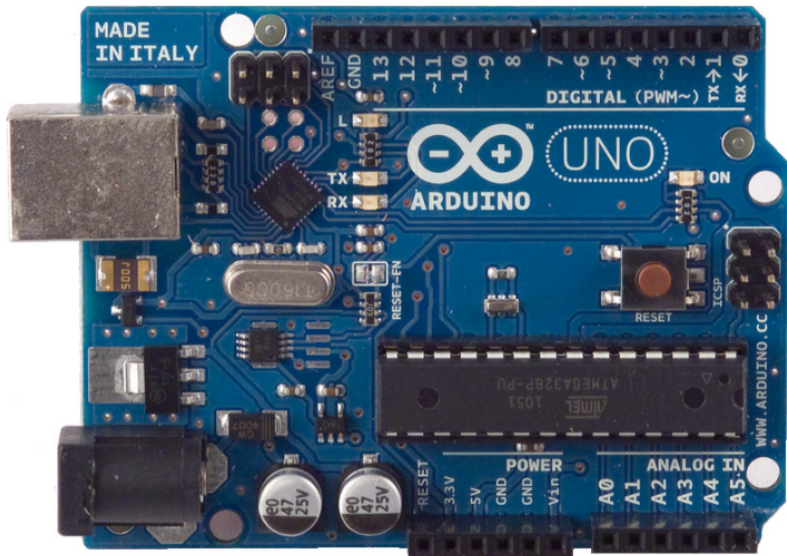
## 3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:



# Microcontrolador

## Arduino UNO



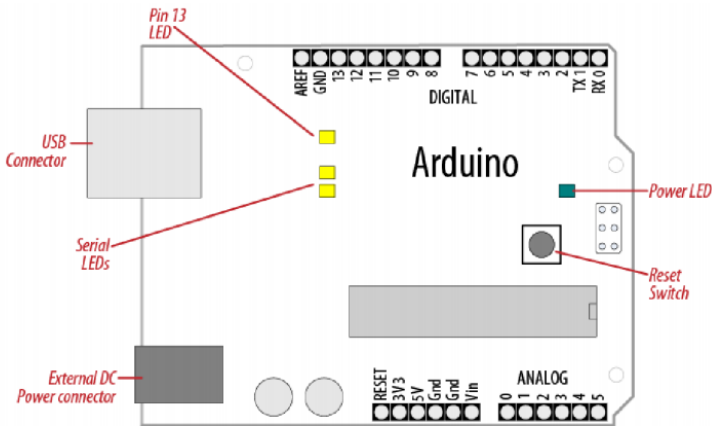
# Microcontrolador

## Características

- Microcontrolador: ATmega328 (**8 bits**).
- Alimentación via USB (5 V) o independiente (7-12 V).
- 14 pines de entrada/salida (I/O) digitales (6 con Pulse Width Modulation: PWM).
- Corriente máxima por pin I/O: **40 mA**.
- 6 pines de entrada analógica.
- Permite comunicación serial.
- Memoria flash: 32 KB.
- Frecuencia del reloj: **16 MHz**.

# Microcontrolador

## Arduino UNO - Placa



M. Margolis, *Arduino Cookbook*, O'Reilly Media 2011.

# Esquema de la presentación

## 1 Introducción

## 2 Hardware

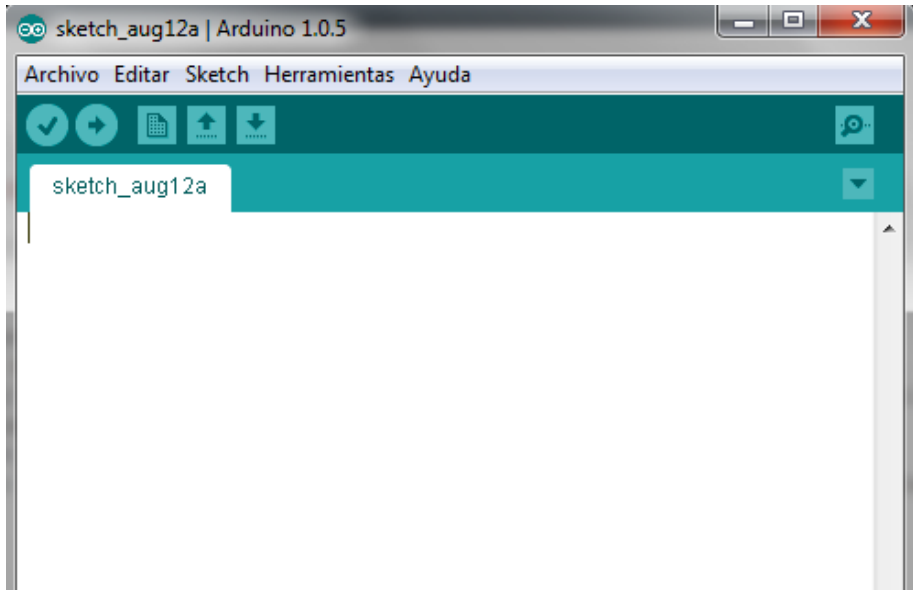
## 3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:

- Basado en Wiring y similar a C++.
- Sketches = Código fuente.
- Lenguaje compilado.

# Entorno de desarrollo: Arduino IDE

-a utilizarse durante la etapa de proyectos-



# Entorno de desarrollo: Arduino IDE

Proceso: Código fuente



The screenshot shows the Arduino IDE interface. The title bar reads "sube\_y\_baja\_brillo\_con\_pote\_comunicacion\_serial | Arduino 1.0.5". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". The toolbar contains icons for a checkmark, a right arrow, a grid, an upload arrow, a download arrow, and a help icon. The sketch name "sube\_y\_baja\_brillo\_con\_pote\_comunicacion\_serial" is displayed in the top right of the editor area. The code in the editor is as follows:

```
const int led = 3;
const int pot = 0;

int brillo;

void setup(){
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  //los pines analogicos se declaran como entradas automaticamente
}

void loop(){
  brillo = analogRead(pot)/4; //entre 0 y 1023
```

# Entorno de desarrollo: Arduino IDE

Proceso: Código fuente -> Compilación ("verify")



```
sube_y_baja_brillo_con_pote_comunicacion_serial
const int led = 3;
const int pot = 0;

int brillo;

void setup(){
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  //los pines analogicos se declaran como entradas automaticamente
}

void loop(){
  brillo = analogRead(pot)/4; //entre 0 y 1023
```



# Entorno de desarrollo: Arduino IDE

Proceso: Código fuente -> Compilación ("verify") -> Programarlo en la placa ("upload")



```
sube_y_baja_brillo_con_pote_comunicacion_serial | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
sube_y_baja_brillo_con_pote_comunicacion_serial
const int led = 3;
const int pot = 0;

int brillo;

void setup(){
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  //los pines analogicos se declaran como entradas automaticamente
}

void loop(){
  brillo = analogRead(pot)/4; //entre 0 y 1023
```

# Simulador on-line: TinkerCad

Comenzar un diseño -asumiendo que cada estudiante ya tiene su cuenta creada-

The screenshot shows the Tinkercad website interface. At the top left is the logo 'TINKER CAD' with 'AUTODESK' above it. To the right are navigation links: 'Galería', 'Blog', 'Aprendizaje', 'Enseñanza', a search icon, and a user profile icon with '56' notifications. On the left is a sidebar menu with a search bar 'Buscar diseños...', 'Diseños 3D', 'Circuits' (highlighted with a yellow arrow), 'Bloques de código' (with a 'NUEVO' badge), 'Lecciones', 'Proyectos' (with a '+ Crear proyecto' button), and 'Tuits' (with a 'Seguir' button). The main content area is titled 'Circuits' and features a green button 'Crear nuevo circuito' (pointed to by a yellow arrow) and a 'Select' checkbox. Below are eight circuit design cards, each with a thumbnail image, a title, a creation date, and a privacy status. The cards are: 'TRtaller4Ej1d' (Público), 'Copy of Ej2' (Privado), 'Super Leelo-Robo' (Privado), 'Magnficent Lappi-Wolt' (Privado), 'L293ybotonesOK' (Privado), 'TRtaller4Ej1a' (Público), 'TR\_T2ej2' (Privado), and 'TR\_T2ej1' (Privado).

# Simulador on-line: TinkerCad

Renombrar el nuevo proyecto, buscar e instanciar el microcontrolador

miProyecto01

Se han guardado todos los cambios.

Código ▶ Iniciar simulación Exportar Compartir

Componentes  
Todos

arduino

Microcontroladores

Arduino Uno R3 ATTiny

# Simulador on-line: TinkerCad

Acceder a la sección de codificación -en general se usará solo *texto*-

miProyecto01

Se han guardado todos los cambios.

Código ▶ Iniciar simulación Exportar Compartir

Bloques + texto

Bloques

- Bloques + texto
- Texto

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

1 (Arduino Uno R3)

definir LED integrado en ALTA

definir pasador 0 en ALTA

definir pasador 3 en 0

girar servo en el pasador 0 a 0

reproducir altavoz en el pasador 0

desactivar el altavoz en pasador 0

imprimir en monitor en serie hello world

definir LED RGB de pasadores 3

Monitor en serie

# Simulador on-line: TinkerCad

Compilar código y condicionalmente empezar la simulación

miProyecto01

Se han guardado todos los cambios.

Código ▶ Iniciar simulación Exportar Compartir

1 (Arduino Uno R3)

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

Bloques + texto

- Salida
- Entrada
- Notación
- Control
- Matemáticas
- Variables

definir LED integrado en ALTA

definir pasador 0 en ALTA

definir pasador 3 en 0

girar servo en el pasador 0 a 0

reproducir altavoz en el pasador 0

desactivar el altavoz en pasador 0

imprimir en monitor en serie hello world

definir LED RGB de pasadores 3

esperar 1 segundos

definir LED integrado en BAJA

esperar 1 segundos

Monitor en serie

# Simulador on-line: TinkerCad

Detener simulación para volver a editar código

miProyecto01

Se han guardado todos los cambios.

Hora de simulador: 00:00:21

Código Detener simulación Exportar Compartir

1 (Arduino Uno R3)

Bloques + texto

- Salida
- Entrada
- Notación
- Control
- Matemáticas
- Variables

definir LED integrado en ALTA

definir pasador 0 en ALTA

definir pasador 3 en 0

girar servo en el pasador 0 a 0

reproducir altavoz en el pasador 0

desactivar el altavoz en pasador 0

imprimir en monitor en serie hello world

definir LED RGB de pasadores 3

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

Monitor en serie

# Simulador on-line: TinkerCad

¿Cómo compartir un proyecto adecuadamente?



AUTODESK  
TINKERCAD

**Circuits**

Crear nuevo circuito

Modificar

miProyecto01  
hace 6 minutos

Opciones

TF  
hai  
Púl



AUTODESK  
TINKERCAD

**Circuits**

Crear nuevo circuito

Modificar

miProyecto01  
hace 6 minutos  
Privado

Propiedades...  
Duplicar  
Mover a proyecto...  
Eliminar

hace 5 meses  
Público

## Privacidad

Público

Viewable and discoverable by everyone



AUTODESK  
TINKERCAD

**Circuits**

Crear nuevo circuito

miProyecto01  
hace 6 minutos  
Público

# Simulador on-line: TinkerCad

¿Cómo compartir un proyecto adecuadamente?

Simular   Añadir imagen

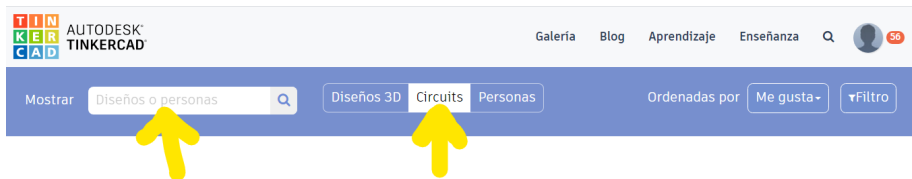
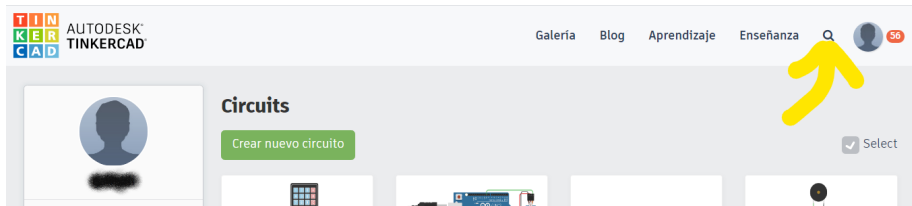
Vínculo: <https://www.tinkercad.com/things/jAdMkxV8is7>

Diseñado por: [User Profile]  
Editado 8/22/20, Creado 8/22/20  
Modificar  
Like 0   Share   Pin it   Incrustar



# Simulador on-line: TinkerCad

¿Cómo buscar un proyecto por su nombre?



# Ejercicios para hacer en el taller

## Ejercicio 1:

Se visitarán distintos proyectos brindados por los docentes (listados en las siguiente slides), con el objetivo que, repartidos en grupos de 3 estudiantes, puedan:

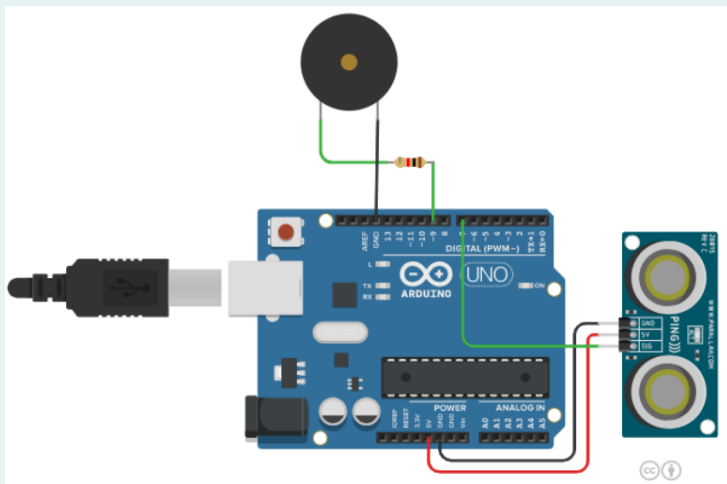
- 1 comenzar a interactuar con el simulador,
- 2 simular el comportamiento y observar la funcionalidad de los distintos circuitos,
- 3 interpretar los códigos, tratando de reconocer la estructura general de éstos y descubriendo la funcionalidad de las distintas instrucciones,
- 4 conocer y clasificar los materiales, interpretando cuáles le dan información al microcontrolador y sobre cuáles el microcontrolador actúa.

A continuación, se presentan los distintos proyectos a visitar. **Para cada uno, buscar en Tinkercad por el nombre del proyecto indicado.**

# Ejercicio 1

## a. Ultrasonido y Buzzer

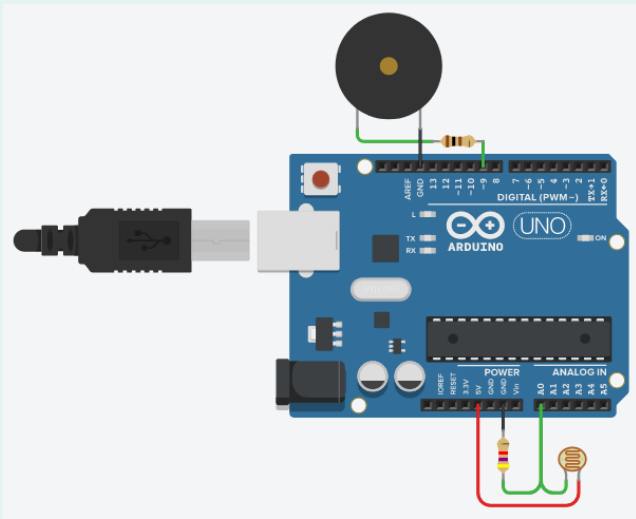
Nombre proyecto: "TRtaller0Ej1a"



# Ejercicio 1

## b. LDR y Buzzer

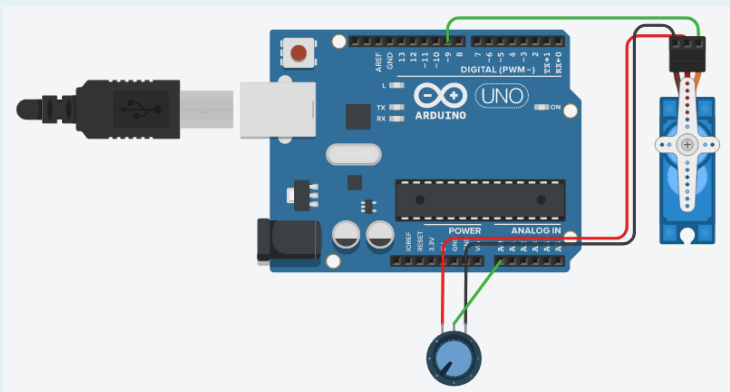
Nombre proyecto: "TRtaller0Ej1b"



# Ejercicio 1

## c. Potenciómetro y Servomotor

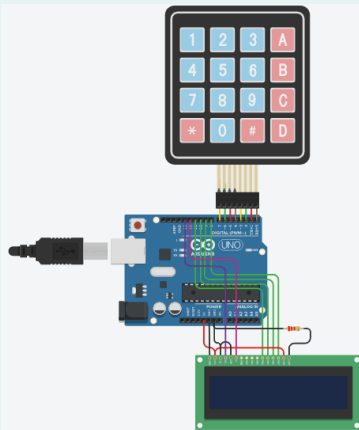
Nombre proyecto: "TRtaller0Ej1c"



# Ejercicio 1

## d. Calculadora

Nombre proyecto: "TRtaller0Ej1d"



# Ejercicio para hacer en casa

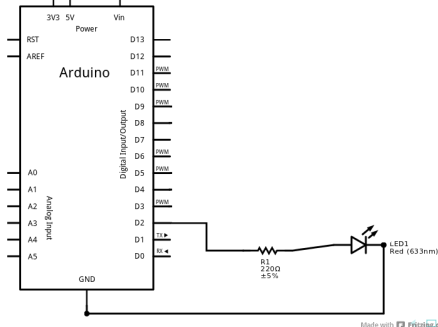
## Ejercicio 1:

Con las conclusiones obtenidas luego de la realización de los ejercicios del taller, se pide:

Implementar en el simulador el circuito de la figura y escribir un programa que prenda y apague un LED cada un segundo.

Para eso utilizar la función `delay(x)` donde `x` es el tiempo que se detiene el programa en milisegundos.

Prestar atención a la estructura del código y a símbolos claves para obtener una compilación "limpia" del código.



## Resumen para la próxima clase:

- 1 Si no se terminaron los ejercicios para hacer en este taller, terminarlos.
- 2 Tener funcionando el Ejercicio 1 de deberes para compartir en la siguiente clase (por dudas utilizar el *Foro de consultas*).
- 3 Queda disponible un cuestionario sobre esta clase, que deberá ser completado en el sitio EVA. **Lo deberá hacer cada estudiante individualmente!!**
- 4 Se recomienda continuar con la lectura de la documentación sugerida en la sección de *Introducción* en el sitio de EVA. En particular, leer el material que podnremos sobre tipos de datos, variables y control de flujo.